周辺分布の高速近似計算と誤り訂正 (符号と暗号の代数的数理)

AUTHOR(S):

渋谷, 智治

CITATION:

渋谷, 智治. 周辺分布の高速近似計算と誤り訂正 (符号と暗号の代数的数理). 数理解析研究所講究録 2005, 1420: 206-215

ISSUE DATE:

2005-04

URL:

http://hdl.handle.net/2433/47177

数理解析研究所講究録 1420 巻 2005 年 206-215

周辺分布の高速近似計算と誤り訂正

渋谷 智治
Tomoharu Shibuya

独立行政法人メディア教育開発センター 研究開発部
R&D Department, National Institute of Multimedia Education

1 まえがき

誤り訂正符号とは, 情報を送信する側において送信情報に冗長を付加することによって, 通信路で生じた誤りを受信側において訂正することを可能にする符号化の技術である. 1990 年代初頭に開発されたターボ符号 [1] や, 1960 年代初頭に提案され [4] その後再発見された low-density parity-check (LDPC) 符号 [7] は誤り訂正符号の一クラスであり, 誤り訂正符号の性能限界 (シャノン限界 [2]) をほぼ達成する符号クラスであることから, 近年多くの研究者の注目を集めている. これらの符号が高い性能を示す鍵は, **belief propagation**[8] や, それと本質的に等価な **sum-product アルゴリズム**[3] に基づく復号法と, sum-product アルゴリズムによる計算の近似精度が向上するような符号構成とにある [7].

信頼性の高い通信を実現するためには, 実際に送信したビットと受信側で推定したビットとが異なる確率 (誤り率) を最小にするような復号戦略をとることが望ましい. このような復号を実現するためには, 受信系列を条件とする送信符号語の条件付確率 (事後確率) を各送信ビットごとに周辺化する手続きが必要となるが, この手続きは一般に符号長の指数関数に比例する計算量を必要とする. このため実際の情報通信でこのような復号法が採用されることはほとんど無く, 上述した周辺化計算をより少ない計算量で精度良く近似するための代替アルゴリズムがこれまでに数多く提案されてきた. その一つである sum-product アルゴリズムは, 符号化の際に織り込まれた送信ビット間の依存関係に注意しながら局所的な和積計算を積み重ねることによって, 符号長に比例する計算量で事後確率の周辺分布をきわめて精度良く近似することが可能である.

本稿では, まず, sum-product アルゴリズムの概要を紹介し, その導出について説明する. また, それと同時に, sum-product アルゴリズムによって計算された周辺分布の近似値が, ある条件下では真の周辺分布に一致することを示す. さらに, sum-product アルゴリズムの応用として, 線形符号の誤り訂正アルゴリズムを紹介する. 最後に, KL ダイバージェンスに密接に関わるある評価関数の極値探索アルゴリズムとして sum-product アルゴリズムが解釈できることを紹介する.

2 誤り訂正符号

本章では, 無記憶通信路上で生じた誤りを線形符号を用いて訂正する原理について説明する. なお, 情報通信における誤り訂正問題の一般的なモデル化については, 情報理論や符号理論の教科書 [2, 6, 9] を参照のこと.

2.1 線形符号

n を自然数とし, $\mathbb{F}_2 = \{0, 1\}$ を二つの要素からなる体とする. $\mathbb{F}_2^n$ の $\mathbb{F}_2$-線形部分空間 $C \subset \mathbb{F}_2^n$ を符号長 n の **2 元線形符号**または単に符号とよぶ.

$\mathbb{F}_2$-線形空間としての C の次元を k とおく．C の基底 $\boldsymbol{g}_1, \boldsymbol{g}_2, \ldots, \boldsymbol{g}_k$ を行とする $k \times n$ 行列:

$$G := \begin{bmatrix} \boldsymbol{g}_1 \\ \boldsymbol{g}_2 \\ \vdots \\ \boldsymbol{g}_k \end{bmatrix}$$

を考えると，$C = \{\boldsymbol{i}G \mid \boldsymbol{i} \in \mathbb{F}_2^k\}$ が成り立つ．このことから，G を符号 C の**生成行列**という．**送信情報** $\boldsymbol{i} \in \mathbb{F}_2^k$ に生成行列 G を掛け合わせることを**符号化**といい，得られた C の要素 $\boldsymbol{i}G$ を送信情報 $\boldsymbol{i}$ に対する**符号語**という．

$C^\perp$ を C に直交する $\mathbb{F}_2^n$ の元の集合:

$$C^\perp := \{\boldsymbol{v} \in \mathbb{F}_2^n \mid \boldsymbol{v} \cdot \boldsymbol{x} = 0 \text{ for all } \boldsymbol{x} \in C\}$$

とする．但し，$\boldsymbol{v} \cdot \boldsymbol{x}$ は $\boldsymbol{v}$ と $\boldsymbol{x}$ の内積を表す．$C^\perp$ を生成する $\boldsymbol{h}_1, \boldsymbol{h}_2, \ldots, \boldsymbol{h}_m \in \mathbb{F}_2^n$ に対して，それらを行とする $m \times n$ 行列

$$H := \begin{bmatrix} \boldsymbol{h}_1 \\ \boldsymbol{h}_2 \\ \vdots \\ \boldsymbol{h}_m \end{bmatrix}$$

を考えると，$\boldsymbol{x} \in \mathbb{F}_2^n$ が符号語であるための必要十分条件は $H\boldsymbol{x}^T = \boldsymbol{0}$ が成り立つこと，即ち，

$$\boldsymbol{h}_\nu \cdot \boldsymbol{x} = 0, \quad \nu = 1, 2, \ldots, m \tag{1}$$

が成り立つことである．このことから，H を符号 C の**検査行列**という．また，k, m を符号 C の**情報点数**, **検査点数**という．一般に $m \geq n - k$ が成り立つ．

2.2 誤り訂正の原理

以下では C を符号とし，送信符号語および受信語をそれぞれ $\boldsymbol{x} = (x_1, x_2, \ldots, x_n) \in C$, $\boldsymbol{y} = (y_1, y_2, \ldots, y_n) \in \mathcal{Y}^n$ で表す．但し $\mathcal{Y}$ は通信路の出力アルファベットを表す．また，$m \times n$ 行列 $H = (h_{\nu i})$ を C の検査行列とし，H に対して

$$\begin{cases} A_\nu := \{i \mid h_{\nu i} = 1\}, & \nu = 1, 2, \ldots, m, \\ B_i := \{\nu \mid h_{\nu i} = 1\}, & i = 1, 2, \ldots, n \end{cases}$$

と定める．

送信符号語は C から一様に選ばれるものとする．$\boldsymbol{x} \in \mathbb{F}_2^n$ が C の符号語であるための必要十分条件(式 (1)) が，A_ν を用いて $\sum_{i \in A_\nu} x_i = 0$ $(\nu = 1, 2, \ldots, m)$ と表せることに注意すると，$\boldsymbol{x} \in \mathbb{F}_2^n$ の事前分布 $P(\boldsymbol{x})$ は

$$P(\boldsymbol{x}) = \frac{1}{|C|} \prod_{\nu=1}^{m} \delta\Big(\sum_{i \in A_\nu} x_i, 0 \Big)$$

と表せる．但し

$$\delta(a, b) := \begin{cases} 1, & \text{if } a = b, \\ 0, & \text{if } a \neq b \end{cases}$$

である．

一方，通信路において誤りが生じる過程は，条件付確率 $P(\boldsymbol{y}|\boldsymbol{x})$ で表される．ここで，$P(\boldsymbol{y}|\boldsymbol{x}) = \prod_{i=1}^{n} P(y_i|x_i)$ が成り立つとき，通信路は**無記憶**であるという．以下では無記憶通信路上で通信が行われるものとする．

以上の仮定の下では, 受信語 $\boldsymbol{y} \in \mathcal{Y}^n$ を得たときの, 送信符号語 $\boldsymbol{x} \in C$ の事後分布 $P(\boldsymbol{x}|\boldsymbol{y})$ は, ベイズの公式より

$$P(\boldsymbol{x}|\boldsymbol{y}) = \frac{P(\boldsymbol{x})P(\boldsymbol{y}|\boldsymbol{x})}{\sum_{\boldsymbol{x}} P(\boldsymbol{x})P(\boldsymbol{y}|\boldsymbol{x})} = \kappa \prod_{\nu=1}^{m} \delta\Big(\sum_{i\in A_\nu} x_i, 0\Big) \prod_{i=1}^{n} P(y_i|x_i) \tag{2}$$

と表される. 但し κ は正規化定数であり, $\sum_{\boldsymbol{x}\in\mathbb{F}_2^n} P(\boldsymbol{x}|\boldsymbol{y}) = 1$ を満たすように定められる.

ここで, 受信側で推定した符号語 $\hat{\boldsymbol{x}} \in C$ が送信符号語 $\boldsymbol{x}$ と異なる確率を最小にするためには, $\hat{\boldsymbol{x}} = (\hat{x}_1, \hat{x}_2, \ldots, \hat{x}_n)$ を

$$\hat{\boldsymbol{x}} = \operatorname*{argmax}_{\boldsymbol{x}\in C} P(\boldsymbol{x}|\boldsymbol{y})$$

で定めればよい [9]. この復号は最大事後確率 (maximum a posteriori probability, MAP) 復号とよばれる[1]. 一方, 推定した各ビット $\hat{x}_i \in \mathbb{F}_2$ が送信ビット x_i と異なる確率を最小にするためには, $\hat{x}_j$ を

$$\hat{x}_i = \operatorname*{argmax}_{x_i\in\mathbb{F}_2} P(x_i|\boldsymbol{y})$$

で定めればよい [9]. ここで, $P(x_i|\boldsymbol{y})$ は, $P(\boldsymbol{x}|\boldsymbol{y})$ の x_i に関する周辺分布

$$P(x_i|\boldsymbol{y}) := \sum_{x_1}\cdots\sum_{x_{i-1}}\sum_{x_{i+1}}\cdots\sum_{x_n} P(\boldsymbol{x}|\boldsymbol{y}) \quad \Big(= \sum_{\boldsymbol{x}\sim x_i} P(\boldsymbol{x}|\boldsymbol{y})\Big) \tag{3}$$

を表す[2]. この復号は周辺事後確率 (maximizer of posterior marginals, MPM) 復号とよばれる [5].

MAP 復号では, C の全ての符号語 (2^k 個) に対して $P(\boldsymbol{x}|\boldsymbol{y})$ を評価し, その最大値を与える符号語 $\boldsymbol{x}$ を探索する必要がある. したがって, k が大きな場合には現実的な時間で MAP 復号を行うことは困難である. 一方 MPM 復号では, $P(\boldsymbol{x}|\boldsymbol{y})$ の周辺化に, 2^{n-1} 個の $\boldsymbol{x} \in \mathbb{F}_2^n$ に対する $P(\boldsymbol{x}|\boldsymbol{y})$ の評価とそれらの和を計算する必要があり, こちらも n が大きい場合に現実的な時間で復号を行うことは困難である.

このように, 誤り訂正における復号問題は計算量的に非常に困難な問題であるといえる. これに対し, sum-product アルゴリズムに基づく反復復号法 [4, 10] (以後, sum-product 復号法と呼ぶ) は, 符号長 n に比例する計算量で MPM 復号を精度良く近似する復号法である.

3 sum-product アルゴリズムと誤り訂正

3.1 sum-product アルゴリズムの概要

離散値をとる確率変数 $x_1, x_2, \ldots, x_6$ の同時分布 $P(\boldsymbol{x}) = P(x_1, x_2, \ldots, x_6)$ が

$$P(\boldsymbol{x}) = f_\alpha(x_1, x_2, x_3) f_\beta(x_1, x_4) f_\gamma(x_4, x_5) f_\delta(x_4, x_6) \tag{4}$$

のように分解できる場合を考える. このとき, 確率変数と因子関数の依存関係を考慮すると, x_1 の周辺分布 $P_1(x_1) := \sum_{\boldsymbol{x}\sim x_1} P(\boldsymbol{x})$ は

$$P_1(x_1) = \underbrace{\Big\{\sum_{x_2}\sum_{x_3} f_\alpha(x_1, x_2, x_3)\Big\}}_{\text{(i)}} \underbrace{\Big\{\sum_{x_4} f_\beta(x_1, x_4) \overbrace{\Big(\underbrace{\sum_{x_5} f_\gamma(x_4, x_5)}_{\text{(ii)}}\Big)\Big(\underbrace{\sum_{x_6} f_\delta(x_4, x_6)}_{\text{(iii)}}\Big)}^{\text{(iv)}}\Big\}}_{\text{(v)}} \tag{5}$$

[1] ここでは $\boldsymbol{x}$ の事前分布を一様と仮定しているので, 最尤 (maximum likelihood, ML) 復号に一致する.
[2] 以後, x_i 以外の全ての変数の和 $\sum_{x_1}\cdots\sum_{x_{i-1}}\sum_{x_{i+1}}\cdots\sum_{x_n}$ を $\sum_{\boldsymbol{x}\sim x_i}$ で表す.

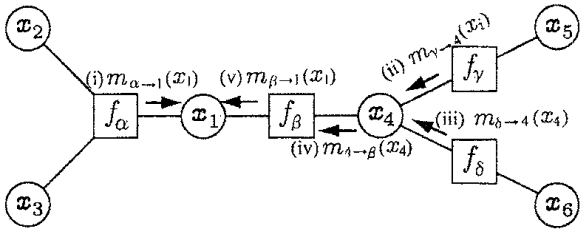

図 1: 式 (4) で与えられる $P(\boldsymbol{x})$ のファクターグラフ. ファクターグラフ上でメッセージが伝播することによって, 周辺分布の計算が実行されると解釈できる.

のように分割して計算できることが示せる. 即ち,

$$\text{[(i), (ii), (iii) の和]} \to \text{[(iv) の積]} \to \text{[(v) の和]} \to \text{[(i) と (v) の積]} \qquad (*)$$

の順番に計算することによって, x_1 の周辺分布 $P_1(x_1)$ を求めることができる. ここで, x_i のとり得る値の個数を n_i で表すと, 式 (3) のような単純な周辺化ではおよそ $n_2n_3n_4n_5n_6$ 回の $P(\boldsymbol{x})$ の評価が $P_1(x_1)$ の計算に必要であるのに対し, 式 (5) ではおよそ $n_2n_3 + n_4(n_5 + n_6)$ 回の因子関数の評価で十分であることがわかる. すなわち, 式 (5) に基づいた周辺分布計算の方が効率的である. このように, 与えられた確率分布を, 局所的な変数集合に対する和とその結果の積に分解して周辺分布計算の高速化を図るのが, sum-product アルゴリズムの基本原理である [3, 8].

一般に $P(\boldsymbol{x})$ $(\boldsymbol{x} = (x_1, x_2, \ldots, x_n))$ が

$$P(\boldsymbol{x}) = \kappa \prod_{\nu=1}^{m} f_\nu(\boldsymbol{x}_\nu) \qquad (6)$$

と因数分解される場合を考える. 但し, $\boldsymbol{x}_\nu$ は f_ν の変数の集合を表す. また, κ は正規化定数である.

式 (6) の分解に対し, x_i, f_ν をラベルとするノードの集合 V, F をそれぞれ $V := \{x_1, x_2, \ldots, x_n\}$, $F := \{f_1, f_2, \ldots, f_m\}$ で定める. 一方, 枝集合 $E \subset V \times F$ を, $x_i \in \boldsymbol{x}_\nu$ かつそのときに限り $(x_i, f_\nu) \in E$ とすることによって定める. このとき, 二部グラフ $\mathcal{G} = (V \cup F, E)$ は, $P(\boldsymbol{x})$ の (式 (6) の因数分解に関する) **ファクターグラフ**(factor graph) と呼ばれる. また V, F の要素をそれぞれ**変数ノード**, **関数ノード**と呼ぶ

図 1 に式 (4) のファクターグラフを示す. この図からわかるように, ファクターグラフを用いることによって, 因数分解における変数と因子関数の関係を視覚化することができる. さらに, 図 1 のようにファクターグラフがループを含まない場合, (*) で記述される式 (5) の計算は, ファクターグラフの端点から開始し, ノードの配列に従って全てのノードに行き渡るまで局所的な和積計算を繰り返すことによって実現されると考えることができる.

ここで, (*) における局所的な和積計算の結果をそれぞれ, ノード f_ν からノード x_i へ送られるメッセージ $(m_{\nu\to i}(x_i))$, ノード x_i からノード f_ν へ送られるメッセージ $(m_{i\to\nu}(x_i))$ であると考える. このとき, (*) で記述される一連の計算は, 図 1 に示すようにあたかもファクターグラフ上をメッセージが伝播することによって実行されているようにみなすことができる.

(*) の和積計算を一般化すると, 次章で示すように, 各ノード間で受け渡しされるメッセージ $m_{\nu\to i}(x_i)$, $m_{i\to\nu}(x_i)$ を計算するアルゴリズム:

$$\left.\begin{aligned} m_{\nu\to i}(x_i) &:= \sum_{\boldsymbol{x}_\nu \sim x_i} f_\nu(\boldsymbol{x}_\nu) \prod_{i' \in N(\nu)\setminus\{i\}} m_{i'\to\nu}(x_{i'}) \\ m_{i\to\nu}(x_i) &:= \prod_{\nu' \in N(i)\setminus\{\nu\}} m_{\nu'\to i}(x_i) \end{aligned}\right\} \qquad (7)$$

が導かれる. さらに, グラフの端点からメッセージを逐次計算し, 全ての $\nu \in N(i)$ に対して $m_{\nu\to i}(x_i)$

が得られた時点で

$$Q_i(x_i) := \kappa_i \prod_{\nu \in N(i)} m_{\nu \to i}(x_i) \tag{8}$$

によって定められる分布を考える．ここで，κ_i は正規化定数，$N(i), N(\nu)$ はファクターグラフにおけるノード x_i に隣接する関数ノードおよびノード f_ν に隣接する変数ノードのインデックスを表す．

ファクターグラフにループがない場合には $Q_i(x_i)$ は $P_i(x_i)$ に一致することが示される．一方，ファクターグラフがループを含む場合には，グラフの端点からメッセージを逐次計算していくと，ある時点で既に計算済みのメッセージを再計算する場合が生じる．つまり，再計算のたびにメッセージの値が変化する可能性があるため，式 (8) の値が定まる保証はなく，従って式 (8) が真の周辺分布を与える保証はない．

しかしながら，ループの長さが比較的大きな場合には，全メッセージを式 (7) に従って同期的に反復更新し，さらにメッセージの収束値を式 (8) に代入して $Q_i(x_i)$ を定めることにより，周辺分布 $P_i(x_i)$ の近似が求まることが期待される．これが，一般の分布に対する sum-product アルゴリズムである．

3.2 sum-product アルゴリズムの導出

本節では，$P(\boldsymbol{x})$ のファクターグラフにループが含まれないとき，式 (7), (8) で記述される sum-product アルゴリズムが与える $Q_i(x_i)$ が x_i の周辺分布 $P_i(x_i)$ に一致することを示す．

$P(\boldsymbol{x})$ が式 (6) のように因数分解されているものとする．$P(\boldsymbol{x})$ のファクターグラフ $\mathcal{G} = (V \cup F, E)$ にループが含まれない場合，$\mathcal{G}$ の変数ノード x_i に対して，$\mathcal{G}$ を x_i を根 (root) とする木とみなすことができる．さらに x_i に対して $\nu \in N(i)$ を一つ定めると，$\mathcal{G}$ にループがないことから，$\mathcal{G}$ の部分グラフで，変数ノード x_i を根とし関数ノード f_ν を含む $\mathcal{G}$ の木が唯一つ定まる．この木を $T(i,\nu)$ で表し，$T(i,\nu)$ に含まれる変数ノードおよび関数ノードのインデックスの集合をそれぞれ $V(i,\nu)$, $F(i,\nu)$ で表す．

容易に確認できるように，全ての変数ノード x_i $(i = 1, 2, \ldots, n)$ に対して

$$\bigcup_{\nu \in N(i)} V(i,\nu) = \{1, 2, \ldots, n\}, \quad \bigcup_{\nu \in N(i)} F(i,\nu) = \{1, 2, \ldots, m\} \tag{9}$$

が成り立つ．また，$N(i)$ の互いに異なる二つの要素 ν_1, ν_2 に対して

$$V(i,\nu_1) \cap V(i,\nu_2) = \emptyset, \quad F(i,\nu_1) \cap F(i,\nu_2) = \{i\} \tag{10}$$

が成り立つ．

命題 1 分布 $P(\boldsymbol{x})$ が式 (6) のように分解されているものとする．

$$m_{\nu \to i}(x_i) := \sum_{x_{i'}: i' \in V(i,\nu) \setminus \{i\}} f_\nu(\boldsymbol{x}_\nu) \prod_{\nu' \in F(i,\nu) \setminus \{\nu\}} f_{\nu'}(\boldsymbol{x}_{\nu'})$$

とおくと，$P(\boldsymbol{x})$ のファクターグラフにループが存在しないとき，$P_i(x_i) = \kappa \prod_{\nu \in N(i)} m_{\nu \to i}(x_i)$ が成り立つ．

(証明) 式 (9), (10) の関係に注意すると，

$$\begin{aligned}
P_i(x_i) &= \kappa \sum_{\boldsymbol{x} \sim x_i} \prod_{\nu=1}^{m} f_\nu(\boldsymbol{x}_\nu) \\
&= \kappa \sum_{\boldsymbol{x} \sim x_i} \prod_{\nu \in N(i)} \prod_{\nu' \in F(i,\nu)} f_{\nu'}(\boldsymbol{x}_{\nu'}) \\
&= \kappa \prod_{\nu \in N(i)} \left(\sum_{x_{i'}: i' \in V(i,\nu) \setminus \{i\}} \prod_{\nu' \in F(i,\nu)} f_{\nu'}(\boldsymbol{x}_{\nu'}) \right) \\
&= \kappa \prod_{\nu \in N(i)} \left(\sum_{x_{i'}: i' \in V(i,\nu) \setminus \{i\}} f_\nu(\boldsymbol{x}_\nu) \prod_{\nu' \in F(i,\nu) \setminus \{\nu\}} f_{\nu'}(\boldsymbol{x}_{\nu'}) \right) \\
&= \kappa \prod_{\nu \in N(i)} m_{\nu \to i}(x_i)
\end{aligned}$$

を得る. □

$m_{\nu\to i}(x_i)$ の定義および命題1から, x_i の周辺分布の計算は, $\mathcal{G}$ の各部分グラフ $T(i,\nu)$ $(\nu\in N(i))$ におけるメッセージ $m_{\nu\to i}(x_i)$ の計算に分割できることがわかる. さらに, 次の命題は, これらのメッセージが $T(i,\nu)$ 内のメッセージのみを用いた漸化式によって表されることを示している.

命題 2 分布 $P(\boldsymbol{x})$ が式 (6) のように分解されているものとする. $P(\boldsymbol{x})$ のファクターグラフにループが存在しないとき

$$m_{\nu\to i}(x_i)=\sum_{\boldsymbol{x}_\nu\sim x_i} f_\nu(\boldsymbol{x}_\nu)\prod_{i'\in N(\nu)\setminus\{i\}}\prod_{\nu'\in N(i')\setminus\{\nu\}} m_{\nu'\to i'}(x_{i'})$$

が成り立つ.

(証明) $V(i,\nu)$, $F(i,\nu)$ の定義より

$$F(i,\nu)\setminus\{\nu\}=\bigcup_{i'\in N(\nu)\setminus\{i\}}\bigcup_{\nu'\in N(i')\setminus\{\nu\}} F(i',\nu')$$

および

$$V(i,\nu)\setminus\{i\}=(N(\nu)\setminus\{i\})\bigcup\left(\bigcup_{i'\in N(\nu)\setminus\{i\}}\bigcup_{\nu'\in N(i')\setminus\{\nu\}} V(i',\nu')\setminus\{i'\}\right)$$

が成り立つ. さらに, 式 (9), (10) の関係に注意すると,

$$\begin{aligned}
m_{\nu\to i}(x_i)&=\sum_{x_{i'}:i'\in V(i,\nu)\setminus\{i\}} f_\nu(\boldsymbol{x}_\nu)\prod_{\nu'\in F(i,\nu)\setminus\{\nu\}} f_{\nu'}(\boldsymbol{x}_{\nu'})\\
&=\sum_{x_{i'}:i'\in V(i,\nu)\setminus\{i\}} f_\nu(\boldsymbol{x}_\nu)\prod_{i'\in N(\nu)\setminus\{i\}}\prod_{\nu'\in N(i')\setminus\{\nu\}}\prod_{\mu\in F(i',\nu')} f_\mu(\boldsymbol{x}_\mu)\\
&=\sum_{\boldsymbol{x}_\nu\sim x_i} f_\nu(\boldsymbol{x}_\nu)\prod_{i'\in N(\nu)\setminus\{i\}}\prod_{\nu'\in N(i)\setminus\{\nu\}}\left(\sum_{x_\ell:\ell\in V(i',\nu')\setminus\{i\}} f_{\nu'}(\boldsymbol{x}_{\nu'})\prod_{\mu\in F(i',\nu')\setminus\{\nu'\}} f_\mu(\boldsymbol{x}_\mu)\right)\\
&=\sum_{\boldsymbol{x}_\nu\sim x_i} f_\nu(\boldsymbol{x}_\nu)\prod_{i'\in N(\nu)\setminus\{i\}}\prod_{\nu'\in N(i')\setminus\{\nu\}} m_{\nu'\to i'}(x_{i'})
\end{aligned}$$

が導かれる. □

ここで $m_{i\to\nu}(x_\nu):=\prod_{\nu'\in N(i)\setminus\{\nu\}} m_{\nu'\to i}(x_i)$ とおくと、命題2より sum-product アルゴリズムの更新式 (7) が直ちに導かれる. また, 命題 1, 2 をあわせると, ファクターグラフの端点から式 (7) に従ってメッセージを更新することにより, 式 (8) によって定まる $Q_i(x_i)$ が真の周辺分布を与えることがわかる.

3.3 sum-product 復号法

2.2節で述べたように, 線形符号の MPM 復号を行うためには, 式 (2) で与えられる事後確率を x_i について周辺化すれば良い. この周辺化を近似的に行うために sum-product アルゴリズムを適用する場合, メッセージの更新式は

$$\left.\begin{aligned}
m_{\nu\to i}(x_i)&:=\sum_{\boldsymbol{x}_\nu:\mathrm{even}\sim x_i}\prod_{i'\in A_\nu\setminus\{i\}} m_{i'\to\nu}(x_{i'})\\
m_{i\to\nu}(x_i)&:=p(y_i|x_i)\prod_{\nu'\in B_i\setminus\{\nu\}} m_{\nu'\to i}(x_i)
\end{aligned}\right\}$$

となる. 但し, $\sum_{\boldsymbol{x}_\nu:\mathrm{even}\sim x_i}$ は, 重みが偶数である $\boldsymbol{x}_\nu$ に対して x_i 以外についての和をとることを意味する. また, $P(x_i|\boldsymbol{y})$ の近似値は

$$Q_i(x_i):=\kappa_i P(y_i|x_i)\prod_{\nu\in B_i} m_{\nu\to i}(x_i) \tag{11}$$

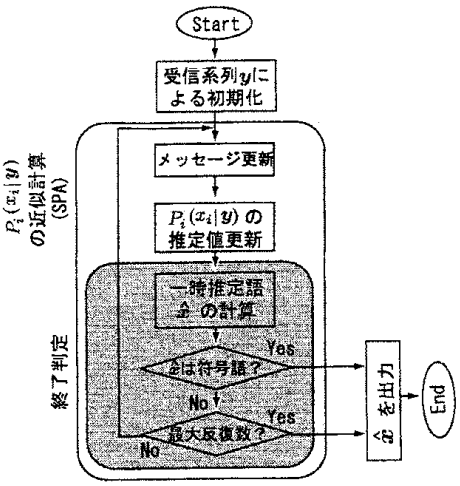

図 2: sum-product 復号アルゴリズム. 受信系列 $\boldsymbol{y}$ に対し, sum-product アルゴリズムを用いて $P_i(x_i|\boldsymbol{y})$ の推定値を計算し, 復号結果 $\hat{\boldsymbol{x}}$ を出力する.

で与えられる. 但し κ_i は正規化定数を表す.

図 2 に線形符号の sum-product 復号アルゴリズムの概要を示す [10]. sum-product 復号では, sum-product アルゴリズムにおいて全メッセージが同期的に更新された後に $P_i(x_i|\boldsymbol{y})$ の推定値を式 (11) で求め,

$$\begin{cases} \hat{\boldsymbol{x}} := (\hat{x}_1, \hat{x}_2, \ldots, \hat{x}_n), \\ \hat{x}_i := \underset{x_i}{\operatorname{argmax}}\{P_i(x_i|\boldsymbol{y})\} \end{cases}$$

により一時推定語 $\hat{\boldsymbol{x}}$ を求める. なお, $H\hat{\boldsymbol{x}} = \boldsymbol{0}$ が成り立つか否かによって, $\hat{\boldsymbol{x}}$ が符号語であるか否かを直ちに判定できる. そこで, sum-product アルゴリズムによるメッセージの反復計算に終了判定を組み込むことができ, 不必要な反復を省くことができる.

4 sum-product アルゴリズムと KL ダイバージェンス

$\boldsymbol{x}$ のある確率分布 $Q(\boldsymbol{x})$ によって確率分布 $P(\boldsymbol{x})$ を近似する場合, 近似の良し悪しを測る尺度として, $Q(\boldsymbol{x})$ と $P(\boldsymbol{x})$ の間の **KL ダイバージェンス**:

$$D(Q||P) := \sum_{\boldsymbol{x}} Q(\boldsymbol{x}) \ln \frac{Q(\boldsymbol{x})}{P(\boldsymbol{x})}$$

がしばしば用いられる. よく知られているように, KL ダイバージェンスは非負であり, $D(Q||P) = 0$ となるための必要十分条件は $Q(\boldsymbol{x}) = P(\boldsymbol{x})$ が成り立つことである. 従って, 与えられた $P(\boldsymbol{x})$ に対して KL ダイバージェンスを小さくする $Q(\boldsymbol{x})$ は, $P(\boldsymbol{x})$ の良好な近似となることが期待される.

補題 3 与えられた分布 $P(\boldsymbol{x})$ が式 (6) のように分解されているものとする. $P(\boldsymbol{x})$ のファクターグラフが木ならば

$$\left.\begin{aligned} P_i(x_i) &:= \sum_{\boldsymbol{x} \sim x_i} P(\boldsymbol{x}) = \kappa \prod_{\nu \in N(i)} m_{\nu \to i}(x_i), \\ P_\nu(\boldsymbol{x}_\nu) &:= \sum_{\boldsymbol{x} \sim \boldsymbol{x}_\nu} P(\boldsymbol{x}) = \kappa f_\nu(\boldsymbol{x}_\nu) \prod_{i \in N(\nu)} \prod_{\nu' \in N(i) \setminus \{\nu\}} m_{\nu' \to i}(x_i) \end{aligned}\right\} \tag{12}$$

が成り立つ.

(証明) 第1式については, sum-product アルゴリズムの結果より明らか. 第2式については

$$\begin{aligned}
P_\nu(\boldsymbol{x}_\nu) &= \kappa \sum_{\boldsymbol{x}\sim\boldsymbol{x}_\nu} \prod_{\nu'=1}^{m} f_{\nu'}(\boldsymbol{x}_{\nu'}) \\
&= \kappa \sum_{\boldsymbol{x}\sim\boldsymbol{x}_\nu} f_\nu(\boldsymbol{x}_\nu) \prod_{\nu'=1,\nu'\neq\nu}^{m} f_{\nu'}(\boldsymbol{x}_{\nu'}) \\
&= \kappa \sum_{\boldsymbol{x}\sim\boldsymbol{x}_\nu} f_\nu(\boldsymbol{x}_\nu) \prod_{i\in N(\nu)} \prod_{\nu'\in N(i)\setminus\{\nu\}} \prod_{\mu\in F(i,\nu')} f_\mu(\boldsymbol{x}_\mu) \\
&= \kappa f_\nu(\boldsymbol{x}_\nu) \prod_{i\in N(\nu)} \prod_{\nu'\in N(i)\setminus\{\nu\}} \left(\sum_{x_{i'}:i'\in V(i,\nu')\setminus\{i\}} f_{\nu'}(\boldsymbol{x}_{\nu'}) \prod_{\mu\in F(i,\nu')\setminus\{\nu'\}} f_\mu(\boldsymbol{x}_\mu) \right) \\
&= \kappa f_\nu(\boldsymbol{x}_\nu) \prod_{i\in N(\nu)} \prod_{\nu'\in N(i)\setminus\{\nu\}} m_{\nu'\to i}(x_i).
\end{aligned}$$

□

命題 4 与えられた分布 $P(\boldsymbol{x})$ が式 (6) のように分解されているものとする. $P(\boldsymbol{x})$ のファクターグラフが木ならば

$$P(\boldsymbol{x}) = \kappa' \frac{\prod_{\nu=1}^{m} P_\nu(\boldsymbol{x}_\nu)}{\prod_{i=1}^{n} P_i^{|N(i)|-1}(x_i)}$$

が成り立つ. 但し, κ' は正規化定数を表す.

(証明)

$$\prod_{\nu=1}^{m} \prod_{i\in N(\nu)} \prod_{\nu'\in N(i)\setminus\{\nu\}} m_{\nu'\to i}(x_i) = \prod_{i=1}^{n} \prod_{\nu\in N(i)} (m_{\nu\to i}(x_i))^{|N(i)|-1}$$

が成り立つことに注意すると, 補題 3 の結果から

$$\frac{\prod_{\nu=1}^{m} P_\nu(\boldsymbol{x}_\nu)}{\prod_{i=1}^{n} P_i^{|N(i)|-1}(x_i)} = \frac{\prod_{\nu=1}^{m} \left(\kappa f_\nu(\boldsymbol{x}_\nu) \prod_{i\in N(\nu)} \prod_{\nu'\in N(i)\setminus\{\nu\}} m_{\nu'\to i}(x_i) \right)}{\prod_{i=1}^{n} \left(\kappa \prod_{\nu\in N(i)} m_{\nu\to i}(x_i) \right)^{|N(i)|-1}} \propto \kappa \prod_{\nu=1}^{n} f_\nu(\boldsymbol{x}_\nu) = P(\boldsymbol{x})$$

が成り立つ. □

ここで, $Q_\nu(\boldsymbol{x}_\nu)$, $Q_i(x_i)$ をそれぞれ $\boldsymbol{x}_\nu$, x_i の分布とし,

$$Q(\boldsymbol{x}) = \kappa' \frac{\prod_{\nu=1}^{m} Q_\nu(\boldsymbol{x}_\nu)}{\prod_{i=1}^{n} Q_i^{|N(i)|-1}(x_i)} \tag{13}$$

で定義される $\boldsymbol{x}$ の分布を考える. 式 (6) で与えられる $P(\boldsymbol{x})$ および式 (13) で与えられる $Q(\boldsymbol{x})$ を $D(Q||P)$ に代入すると

$$D(Q||P) := \sum_{\boldsymbol{x}} Q(\boldsymbol{x}) \sum_{\nu=1}^{m} \log \frac{Q_\nu(\boldsymbol{x}_\nu)}{f_\nu(\boldsymbol{x}_\nu)} - \sum_{\boldsymbol{x}} Q(\boldsymbol{x}) \sum_{i=1}^{n} (|N(i)|-1) Q_i(x_i) \log Q_i(x_i) + c \tag{14}$$

を得る. 但し c は定数. ここで, $P(\boldsymbol{x})$ のファクターグラフが木であるとき, 命題 4 より, 式 (13) で与えられる $Q(\boldsymbol{x})$ で $Q(\boldsymbol{x}) = P(\boldsymbol{x})$ となるものが存在する. さらに, 補題 3 より, このような $Q(\boldsymbol{x})$ に関して $Q_\nu(\boldsymbol{x}_\nu) = \sum_{\boldsymbol{x}\sim\boldsymbol{x}_\nu} Q(\boldsymbol{x})$, $Q_i(x_i) = \sum_{\boldsymbol{x}\sim x_i} Q(\boldsymbol{x})$ が成り立っている. これらの関係を式 (14) に代入すると

$$D(Q||P) = \sum_{\nu=1}^{m} \sum_{\boldsymbol{x}_\nu} Q_\nu(\boldsymbol{x}_\nu) \log \frac{Q_\nu(\boldsymbol{x}_\nu)}{f_\nu(\boldsymbol{x}_\nu)} - \sum_{i} (|N(i)|-1) \sum_{x_i} Q_i(x_i) \log Q_i(x_i) + c$$

となる. ここで

$$D^*(Q||P) := \sum_{\nu=1}^{m}\sum_{\boldsymbol{x}_\nu} Q_\nu(\boldsymbol{x}_\nu)\log\frac{Q_\nu(\boldsymbol{x}_\nu)}{f_\nu(\boldsymbol{x}_\nu)} - \sum_i (|N(i)|-1)\sum_{x_i} Q_i(x_i)\log Q_i(x_i)$$

とおくと, KL ダイバージェンスの性質から, $D(Q||P)^*$ を最小化する分布 $\{Q_\nu(\boldsymbol{x}_\nu), Q_i(x_i)\}$ を求めることによって, 所望の周辺分布 $P_i(x_i)$ が求められる. 但し, $\{Q_\nu(\boldsymbol{x}_\nu), Q_i(x_i)\}$ は, それらが分布となるよう,

$$\sum_{\boldsymbol{x}_\nu} Q_\nu(\boldsymbol{x}_\nu) = 1, \quad \sum_{x_i} Q_i(x_i) = 1 \tag{15}$$

を満たさなければならない. さらに, $\boldsymbol{x}_\nu$ に x_i が含まれる場合,

$$\sum_{\boldsymbol{x}_\nu \sim x_i} Q_\nu(\boldsymbol{x}_\nu) = Q_i(x_i) \tag{16}$$

が成り立つ必要がある.

式 (15), (16) の制約の下で $D^*(Q||P)$ を最小化する $\{Q_\nu(\boldsymbol{x}_\nu), Q_i(x_i)\}$ は, $D^*(Q||P)$ が極値をとるための必要条件, すなわち, Lagrange 乗数形式:

$$L := D^*(Q||P) + \sum_{i=1}^{n}\gamma_i\left(1 - \sum_{x_i} Q_i(x_i)\right) + \sum_{\nu=1}^{m}\sum_{i\in N(\nu)}\sum_{x_i}\lambda_{\nu i}(x_i)\left(Q_i(x_i) - \sum_{\boldsymbol{x}_\nu\setminus x_i} Q_\nu(\boldsymbol{x}_\nu)\right) \tag{17}$$

において $\frac{\partial L}{\partial Q_i(x_i)} = 0$, $\frac{\partial L}{\partial Q_\nu(\boldsymbol{x}_\nu)} = 0$ として得られる連立方程式の解として得られることが多い. ここで, γ_i, $\lambda_{\nu i}(x_i)$ は, 式 (15), (16) で与えられる各制約に対応する Lagrange 未定乗数である. なお, 式 (15) の 1 番目の制約は, 2 番目の制約および式 (16) の制約から得られるため省略している. 実際にこれらの連立方程式を解くと

$$\left.\begin{aligned} Q_i(x_i) &= \exp\left[\tfrac{1}{|N(i)|-1}\left(\textstyle\sum_{\nu\in N(i)}\lambda_{\nu i}(x_i) - \gamma_i\right) - 1\right], \\ Q_\nu(\boldsymbol{x}_\nu) &= f_\nu(\boldsymbol{x}_\nu)\exp\left[\textstyle\sum_{i\in N(\nu)}\lambda_{\nu i}(x_i) - 1\right] \end{aligned}\right\} \tag{18}$$

を得る.

$P(\boldsymbol{x})$ のファクターグラフが木であるとき, 式 (12), (18) において $P_i(x_i) = Q_i(x_i)$, $P_\nu(\boldsymbol{x}_\nu) = Q_\nu(\boldsymbol{x}_\nu)$ が成り立っていることに注意すると, sum-product アルゴリズムによって求められたメッセージ $m_{\nu\to i}(x_i)$, $m_{i\to\nu}(x_\nu)$ および極値条件を満たす Lagrange 未定乗数 $\lambda_{\nu i}(x_i)$ を,

$$\lambda_{\nu i} = \sum_{\nu'\in N(i)\setminus\{\nu\}} \ln m_{\nu'\to i}(x_i) \quad (= \ln m_{i\to\nu}(x_\nu)) \tag{19}$$

によって対応付けることができる [11]. 実際, 式 (19) を式 (18) に代入して整理すると

$$\begin{aligned} Q_i(x_i) &= \exp\left[-\frac{\gamma_i}{|N(i)|-1} - 1\right]\exp\left[\frac{1}{|N(i)|-1}\sum_{\nu\in N(i)}\sum_{\nu'\in N(i)\setminus\{\nu\}}\ln m_{\nu'\to i}(x_i)\right] \\ &= \exp\left[-\frac{\gamma_i}{|N(i)|-1} - 1\right]\exp\left[\frac{1}{|N(i)|-1}\sum_{\nu\in N(i)}(|N(i)|-1)\ln m_{\nu\to i}(x_i)\right] \\ &= \exp\left[-\frac{\gamma_i}{|N(i)|-1} - 1\right]\prod_{\nu\in N(i)} m_{\nu\to i}(x_i) \end{aligned}$$

となり, γ_i を適当に定めることによって $Q_i(x_i) = P_i(x_i)$ が確かめられる. 同様に

$$Q_\nu(\boldsymbol{x}_\nu) = f_\nu(\boldsymbol{x}_\nu)\exp\left[\sum_{i\in N(\nu)}\sum_{\nu'\in N(i)\setminus\{\nu\}} m_{\nu'\to i}(x_i) - 1\right] = e^{-1}f_\nu(\boldsymbol{x}_\nu)\prod_{i\in N(\nu)}\prod_{\nu'\in N(i)\setminus\{\nu\}} m_{\nu'\to i}(x_i)$$

より $Q_\nu(\boldsymbol{x}_\nu) = P_\nu(\boldsymbol{x}_\nu)$ が確かめられる. 以上の考察から, sum-product アルゴリズムのメッセージ更新は, 極値条件を満たす Lagrange 未定乗数の探索アルゴリズムとみなすことができる.

ここで, 式 $Q_\nu(\boldsymbol{x}_\nu)$, $Q_i(x_i)$ を, ($\boldsymbol{x}_\nu, x_i$ の $Q(\boldsymbol{x})$ に対する周辺分布とは限らない) 一般の $\boldsymbol{x}_\nu, x_i$ の分布とおくと, 式 (17) で定義される L は $P(\boldsymbol{x})$ のファクターグラフにループがあるか否かにかかわらず定義可能な量であることがわかる. 但し, この場合には $\sum_{\boldsymbol{x}\sim\boldsymbol{x}_\nu} Q(\boldsymbol{x}) = Q_\nu(\boldsymbol{x}_\nu)$, $\sum_{\boldsymbol{x}\sim x_i} Q(\boldsymbol{x}) = Q_i(x_i)$ を満たす $Q(\boldsymbol{x})$ の存在は保証されず, $D^*(Q||P)$ が KL ダイバージェンスとなるとは限らない. しかしながら, $P(\boldsymbol{x})$ のファクターグラフ内のループの長さが大きければ, $D^*(Q||P)$ は KL ダイバージェンスの近似を与えていることが期待でき, L の極値条件から周辺分布の近似値を得ることができる.

5 むすび

本稿では, 周辺分布の高速近似計算アルゴリズムである sum-product アルゴリズムを紹介し, ファクターグラフが木である場合には, アルゴリズムの結果が真の周辺分布に一致することを示した. また, sum-product アルゴリズムの応用として, 線形符号の MPM 復号問題を近似的に実現する sum-product 復号を紹介した. さらに, ふたつの分布の近さの尺度としてしばしば用いられる KL ダイバージェンスに密接に関わるある評価関数を導入すると, sum-product アルゴリズムが収束した際のメッセージとその評価関数の勾配が零となる点が一対一に対応することを見た.

参考文献

[1] C. Berrou, A. Glavieux, and P. Thitimajshima, "Near Shannon limit error-correcting coding and decoding: Turbo-codes," *Proc. of ICC* (Geneva), pp.1064–1070, 1993.

[2] T. M. Cover and J. A. Thomas, *Elements of Information Theory*, Wiley Series in Telecommunications, Wiley, 1991.

[3] B. J. Frey, *Graphical Models for Machine Learning and Digital Communication*, The MIT Press, 1998.

[4] R. G. Gallager, "Low-density parity-check codes," *IRE Trans. Inform. Theory* vol.IT-8, pp.21–28, 1962.

[5] 汪 金芳, 田栗 正章, 手塚 集, 樺島 祥介, 上田 修功, 計算統計 I, 確率計算の新しい手法 (統計科学のフロンティア 11), 岩波書店, 2003.

[6] J. H. van Lint, *Introduction to Coding Theory*. Springer-Verlag, second ed., 1991.

[7] D. J. C. MacKay, "Good error-correcting codes based on very sparse matrices," *IEEE Trans. Inform. Theory*, vol.IT-45, pp.399–431, 1999.

[8] J. Pearl, *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference*, Morgan Kaufmann Publishers, 1988.

[9] J. G. Proakis, *Digital Communications*, 3rd. ed., McGraw-Hill, 1995.

[10] 和田山正, 低密度パリティ検査符号とその復号法, トリケップス, 2002.

[11] J. S. Yedidia, W. T. Freeman, and Y. Weiss, "Bethe free energy, Kikuchi approximations, and belief propagation algorithms," *Tech. Rep. of Mitsubishi Electric Research Lab.*, TR-2001-16, 2001.